

Robust Multi-Agent Optimization: Coping with Packet-Dropping Link Failures

Lili Su, *Student Member, IEEE*, and Nitin H. Vaidya, *Fellow, IEEE*

Abstract—We study the problem of multi-agent optimization in the presence of communication failures, where agents are connected by a strongly connected communication network. Specifically, we are interested in optimizing $h(x) = \frac{1}{n} \sum_{i=1}^n h_i(x)$, where $\mathcal{V} = \{1, \dots, n\}$ is the set of agents, and $h_i(\cdot)$ is agent i 's local cost function. We consider the scenario where the communication links may suffer packet-dropping failures (i.e., the sent messages are not guaranteed to be delivered in the same iteration), but each link is reliable at least once in every B consecutive message transmissions. This bounded time reliability assumption is reasonable since it has been shown that with unbounded message delays, convergence is not guaranteed for reaching consensus [2] – a special case of the optimization problem of interest. We propose a robust distributed optimization algorithm wherein each agent updates its local estimate using slightly different routines in odd and even iterations. We show that these local estimates converge to a common optimum of $h(\cdot)$ sub-linearly at convergence rate $O(\frac{1}{\sqrt{t}})$, where t is the number of iteration. Our proposed algorithm combines the Push-Sum Distributed Dual Averaging method [24] with a robust average consensus algorithm [26]. The main analysis challenges come from the fact that the effective communication network is time varying, and that each agent does not know the actual number of reliable outgoing links at each iteration.

Index Terms—Optimization, multi-agent systems, fault-tolerance, communication failures; dual averaging method.

1 INTRODUCTION

NETWORKED multi-agent systems consist of a group of agents that collectively perform collaborative tasks. We consider the problem of optimizing the additive cost over multi-agent networks, where the additive cost refers to the average of all the local cost functions associated with individual computing agents [7], [14], [16], [24]. Additive cost objectives appear frequently in practice, such as the regression and classification problems in machine learning [1].

Multi-agent networks are subject to failures. In this work, we focus on the scenario where the communication links may suffer packet-dropping failures (i.e., the sent messages are not guaranteed to be delivered in that iteration), and when the failure occurs, the sender is not immediately aware of this message loss. This scenario arises frequently in wireless networks when agents communicate with each other via message broadcasting. Due to the existence of noise and interruption during message transmission, an agent may not receive the messages sent by its incoming neighbors. Although acknowledgement mechanisms can be incorporated to improve reliability, this may slow down the convergence due to the need for acknowledgement retransmission (requiring more time for each iteration of the algorithm).

A special case of the distributed optimization problem concerns the computation of the exact average of the agents' initial inputs – with each local function being constant. Consensus is a canonical problem in cooperative control [2], [4], [11], [15], [22], [23]. Achieving robust average consensus

over unreliable links has received significant attention [6], [8], [9], [19], [26]. Undirected graphs were considered in [8], [19], where the link failures affect the communication in both directions. In particular, dynamically changing data and networks are considered in [8]. Directed graphs were first considered in [9], however, only biased average was achieved. This bias was later corrected in [6], [26] via introducing auxiliary variables at each agent. Our proposed algorithm uses the robust average algorithms proposed in [8], [26] as algorithmic primitives.

There has been significant research on the additive cost minimization problem [5], [7], [13], [14], [16], [17], [20], [21], [24], [25]. The need for robustness for distributed optimization problems in the presence of link failures also has received some attentions recently [7], [13], [14]. In particular, Duchi et al. [7] assume that each realizable link failure pattern admits a doubly-stochastic matrix which governs the evolution dynamics of local estimates of the optimum. In the case agents know the number of reliable outgoing links at each iteration [14], the requirement for the doubly stochastic matrices is removed using push-sum. Note that link failures may result in message delays when the dropped messages can be recovered once the link is reliable. However, the resulting message delays are dependent across iterations in the sense that in each iteration, all the temperately delayed messages can jointly either be delivered or remain to be buffered. Distributed optimization under communication delays is considered in [25], where random time varying bounded delays are considered. In particular, the random message delays in [25] are independently distributed across agents and across iterations. Our algorithm builds upon the Push-Sum Distributed Dual Averaging method proposed in [24], where the network is assumed to be static. In contrast, in the presence of link failures, the effective communication

• L. Su and N. H. Vaidya are with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, 61801.
E-mail: {lilisu3, nhv}@illinois.edu

network is time varying. Although it is claimed in [24] that their results can be adapted to time varying networks, no formal proofs are presented. In addition, the results in [24] do not apply to the scenario when links are prone to failures.

In this paper, we propose a robust distributed optimization algorithm which combines the Push-Sum Distributed Dual Averaging method [24] with a robust average consensus algorithm [26]. In particular, in our proposed algorithm each agent updates its local estimate using slightly different routines in odd and even iterations. We show that the local estimates converge to a common optimum sub-linearly at convergence rate $O(\frac{1}{\sqrt{t}})$, where t is the number of iteration. The main analysis challenges come from the fact that the effective communication network is time varying, and that each agent does not know the actual number of reliable outgoing links at each iteration.

The rest of this paper is organized as follows. Section 2 presents our system model and problem formulation. In Section 3, we give a detailed description of our algorithm – Robust Push-Sum Distributed Dual Averaging (RPSDA), and provide a matrix representation of the state evolution under RPSDA. The convergence of RPSDA is proved in Section 4. Section 5 concludes the paper.

2 SYSTEM MODEL AND PROBLEM FORMULATION

The system under consideration is synchronous – there is an upper bound on agents' relative speeds and on message delays. If a message is not delivered within the specified time interval, then we say that this message is dropped during transmission. The system consists of n agents that are connected by a *strongly connected* network $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, n\}$ is the collection of agents, and \mathcal{E} is the collection of *directed* communication links among agents. In addition, we assume that each agent can send message to itself, i.e., there is a self-loop $(i, i) \in \mathcal{E}$ at each agent $i \in \mathcal{V}$. Define $\mathcal{I}_i = \{j \mid (j, i) \in \mathcal{E}\}$ and $\mathcal{O}_i = \{j \mid (i, j) \in \mathcal{E}\}$ to be the collections of incoming neighbors and outgoing neighbors, respectively, of agent i . Note that $i \in \mathcal{I}_i$ and $i \in \mathcal{O}_i$. Let $d_i^{\text{out}} = |\mathcal{O}_i|$ be the outdegree of agent i .

In this paper, we focus on the scenario where links are only guaranteed to be reliable at least once every B transmissions. Specifically, at each message transmission, the link $(i, j) \in \mathcal{E}$ may drop the message sent from agent i to agent j ; however, within any B consecutive message transmissions, link (i, j) is reliable at least once. This bounded time reliability assumption is reasonable since it has been shown that with unbounded message delays, convergence is not guaranteed for reaching consensus [2] – a special case of the optimization problem of interest. Note that self-loops are always reliable (unlike the other links). Similar assumptions are adopted in [14], [16]. In addition, due to the potential message dropping, the effective communication network is time varying.

Each agent i knows a local cost function $h_i(\cdot) : \mathcal{X} \rightarrow \mathbb{R}$ that is convex and L -Lipschitz continuous with respect to ℓ_2 norm, i.e.,

$$\|h_i(x) - h_i(y)\|_2 \leq L \|x - y\|_2, \forall x, y \in \mathcal{X}, \forall i \in \mathcal{V}, \quad (1)$$

where $\mathcal{X} \subseteq \mathbb{R}^d$ is nonempty, convex and compact. One immediate consequence of (1) is that $\|g_i\|_2 \leq L$ for any

$x \in \mathcal{X}$ and any subgradient $g_i \in \partial h_i(x)$. To simplify notation, we drop the subscript for ℓ_2 norm.

Our goal is to design an algorithm under which all the agents collaboratively minimize the average cost over the network, i.e., collaboratively compute

$$\arg \min_{x \in \mathcal{X}} h(x) \triangleq \frac{1}{n} \sum_{i=1}^n h_i(x). \quad (2)$$

Let X^* be the collection of optimal solutions of $h(\cdot)$ subject to \mathcal{X} . Since $\mathcal{X} \subseteq \mathbb{R}^d$ is a nonempty, convex and compact, X^* is also nonempty, convex and compact.

Throughout this paper, we use the terms agent and node interchangeably, link and edge interchangeably, as well as packet and message interchangeably.

3 ROBUST PUSH-SUM DISTRIBUTED DUAL AVERAGING

In this section, we provide a formal description of our algorithm that is robust to packet-dropping link failures. To make the paper self-contained, we review some relevant background on dual averaging method. Readers can consult [18] for complete derivation and analysis.

3.1 Dual Averaging Method

Dual averaging method is proposed by Nesterov in [18] for the following nonsmooth convex minimization problem

$$\min_{x \in \mathbb{R}^d} h(x), \quad (3)$$

where $h(\cdot)$ is nonsmooth and convex. For $t \geq 0$, one typical iterate of a subgradient method is as follows,

$$x[t+1] = x[t] - \alpha[t]g[t], \quad (4)$$

where $g[t] \in \partial h(x[t])$ is a subgradient of function $h(\cdot)$ at point $x[t]$, and $\alpha[t]$ is the stepsize at time t . Since $h(\cdot)$ is nonsmooth, the subgradients may not be vanishing in the neighborhood of the optimal solution(s) of (3). Thus, for the subgradient update (4) to converge, the stepsizes $\alpha[t]$ need to be diminishing, i.e., $\alpha[t] \rightarrow 0$. Informally speaking, with diminishing stepsizes, new subgradient (i.e., $g[t]$) is less incorporated into $x[t+1]$ compared to $g[r]$ for $r < t$. This is counter-intuitive, since in order to have x learn an optimal solution x^* , new information should be at least as important as old information.

The above contradiction on the choice of stepsizes motivates the proposition of the dual averaging method, in which, in addition to the estimate sequence $\{x[t]\}_{t=0}^\infty$, there is an additional sequence $\{z[t]\}_{t=0}^\infty$ in the dual space that essentially aggregates all the subgradients generated so far. In addition, the dual averaging scheme involves a *proximal function* $\psi(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$ that is strongly convex. We choose $\psi(\cdot)$ to be 1-strongly convex with respect to ℓ_2 , i.e.,

$$\psi(y) \geq \psi(x) + \langle \Delta\psi(x), y - x \rangle + \frac{1}{2} \|x - y\|^2, \forall x, y \in \mathbb{R}^d.$$

In addition, we assume that $\psi(\cdot)$ to be nonnegative and $\arg \min_x \psi(x) = \mathbf{0} \in \mathbb{R}^d$, which is also referred as proximal center. This choice of $\psi(\cdot)$ is rather standard [7], [24]. As it can be seen later, this proximal function can be used to smooth the update of the primal sequence $\{x[t]\}_{t=0}^\infty$.

One typical iterate of dual averaging method is as follows. Let $z[0] = x[0] = \mathbf{0} \in \mathbb{R}^d$. For $t \geq 0$, compute $g[t] \in \partial h(x[t])$, and update z and x as

$$z[t+1] = z[t] + g[t], \quad (5)$$

and

$$x[t+1] = \prod_{x \in \mathbb{R}^d}^{\psi} (z[t+1], \alpha[t]), \quad (6)$$

where $\prod_{x \in \mathbb{R}^d}^{\psi}(\cdot)$ is the projection operator defined as

$$\prod_{x \in \mathbb{R}^d}^{\psi} (z, \alpha) \triangleq \operatorname{argmin}_{x \in \mathbb{R}^d} \left\{ \langle z, x \rangle + \frac{1}{\alpha} \psi(x) \right\}. \quad (7)$$

From (5) and (6), we know that the update of x is based on all the subgradients generated so far, and all these subgradients are weighted equally.

The convergence rate of the dual averaging method is $O(1/\sqrt{t})$, which is faster than the subgradient method (whose convergence rate is $O(\log t/\sqrt{t})$). Besides, the constants of the dual averaging method are often smaller [18].

Distributed dual averaging is first investigated by Duchi et al. in [7], where individual agents repeatedly exchange their local dual sequence $\{z_i[t]\}_{t=1}^{\infty}$ with their neighbors such that

$$\lim_{t \rightarrow \infty} z_i[t+1] = \frac{1}{n} \lim_{t \rightarrow \infty} \sum_{r=0}^t \sum_{i=1}^n g_i[r],$$

where $g_i[r]$ is a subgradient of $h_i(\cdot)$ at $x_i[r]$, i.e., $g_i[r] \in \partial h_i(x_i[r])$. Doubly stochastic matrices are assumed in [7], and relaxed later in [24] for static networks [12].

3.2 Robust Push-Sum Distributed Dual Averaging

In this subsection, we present our algorithm, named *Robust Push-Sum Distributed Dual Averaging* (RPSDA). In our RPSDA, each agent i locally keeps (1) estimate sequence $\{x_i[t]\}_{t=0}^{\infty}$, (2) gradient aggregation sequence $\{z_i[t]\}_{t=0}^{\infty}$, and (3) weight sequence $\{w_i[t]\}_{t=0}^{\infty}$, where $x_i[0] = z_i[0] = \mathbf{0} \in \mathbb{R}^d$ and $w_i[0] = 1 \in \mathbb{R}$. The weight sequence $\{w_i[t]\}_{t=0}^{\infty}$ is introduced due to the adoption of push-sum mechanism, whose correctness relies crucially on the mass preservation property [3], [12]. For this property to hold in the presence of link failures, in addition to the aforementioned three sequences, each agent needs to keep track of the total “mass” that it has *sent* to each of its *outgoing* neighbors, and the total “mass” that it has *received* from each of its *incoming* neighbors. For $t \geq 0$, let $\sigma_i[t]$ and $\tilde{\sigma}_i[t]$ be the total weighted gradients and weights (up to iteration t) that agent i wants to *transfer* to each of its outgoing neighbors, with $\sigma_i[0] = \mathbf{0} \in \mathbb{R}^d$, $\tilde{\sigma}_i[0] = 0 \in \mathbb{R}$; and for each $j \in \mathcal{I}_i$, let $\rho_{ji}[t]$ and $\tilde{\rho}_{ji}[t]$ be the total weighted gradients and weights (up to iteration t) that agent i has *received* from its incoming neighbor j , with $\rho_{ji}[0] = \mathbf{0} \in \mathbb{R}^d$, $\tilde{\rho}_{ji}[0] = 0 \in \mathbb{R}$. The pseudo-code, listed in Algorithm 1 and Algorithm 2, describes the steps that should be performed by each agent i . The steps are slightly different in odd and even iterations.

No inter-agent communication is involved in odd iterations. Indeed, the operations in odd iterations can be viewed as the special instance of even iterations when all links (except for self-loops) are unreliable. Note that subgradient is added to $z_i[t]$ only at even iterations, with odd time

indices. For ease of exposition, when t is even, we define $g_i[t] = \mathbf{0} \in \mathbb{R}^d$ for $i \in \mathcal{V}$. It is easy to see that the update of x_i is mainly performed in Algorithm 1. Indeed, the odd iterations are introduced due to the technical issues arise in the correctness analysis of RPSDA.

Algorithm 1: RPSDA (even iterations $2t, t \geq 1$)

- 1 *Initialization*: $z_i[0] = x_i[0] = \sigma_i[0] = \mathbf{0} \in \mathbb{R}^d$, $\tilde{\sigma}_i[0] = 0 \in \mathbb{R}$, $w_i[0] = 1 \in \mathbb{R}$, $\rho_{ji}[0] = \mathbf{0} \in \mathbb{R}^d$ and $\tilde{\rho}_{ji}[0] = 0 \in \mathbb{R}$ for each $j \in \mathcal{I}_i$;
 - 2 $\sigma_i[2t] \leftarrow \sigma_i[2t-1] + \frac{z_i[2t-1]}{d_i^{out}}$, $\tilde{\sigma}_i[2t] \leftarrow \tilde{\sigma}_i[2t-1] + \frac{w_i[2t-1]}{d_i^{out}}$, and broadcast $(\sigma_i[2t], \tilde{\sigma}_i[2t])$ to agents in \mathcal{O}_i ;
 - 3 For each $j \in \mathcal{I}_i$: **if** message $(\sigma_j[2t], \tilde{\sigma}_j[2t])$ is received **then**
 - 4 | $\rho_{ji}[2t] \leftarrow \sigma_j[2t]$, $\tilde{\rho}_{ji}[2t] \leftarrow \tilde{\sigma}_j[2t]$;
 - 5 **else**
 - 6 | $\rho_{ji}[2t] \leftarrow \rho_{ji}[2t-1]$, $\tilde{\rho}_{ji}[2t] \leftarrow \tilde{\rho}_{ji}[2t-1]$;
 - 7 **end**
 - 8 Compute a subgradient $g_i[2t-1] \in \partial h_i(x_i[2t-1])$, and $z_i[2t] \leftarrow \sum_{j \in \mathcal{I}_i} (\rho_{ji}[2t] - \rho_{ji}[2t-1]) + g_i[2t-1]$, $w_i[2t] \leftarrow \sum_{j \in \mathcal{I}_i} (\tilde{\rho}_{ji}[2t] - \tilde{\rho}_{ji}[2t-1])$;
 - 9 $x_i[2t] \leftarrow \prod_{\mathcal{X}}^{\psi} \left(\frac{z_i[2t]}{w_i[2t]}, \alpha[2t-1] \right)$;
-

Algorithm 2: RPSDA (odd iterations $2t-1, t \geq 1$)

- 1 $\sigma_i[2t-1] \leftarrow \sigma_i[2t-2] + \frac{z_i[2t-2]}{d_i^{out}}$, $\tilde{\sigma}_i[2t-1] \leftarrow \tilde{\sigma}_i[2t-2] + \frac{w_i[2t-2]}{d_i^{out}}$;
 - 2 For each $j \in \mathcal{I}_i$ such that $j \neq i$:
 $\rho_{ji}[2t-1] \leftarrow \rho_{ji}[2t-2]$, $\tilde{\rho}_{ji}[2t-1] \leftarrow \tilde{\rho}_{ji}[2t-2]$, and $\rho_{ii}[2t-1] \leftarrow \sigma_i[2t-1]$, $\tilde{\rho}_{ii}[2t-1] \leftarrow \tilde{\sigma}_i[2t-1]$;
 - 3 $z_i[2t-1] \leftarrow \sum_{j \in \mathcal{I}_i} (\rho_{ji}[2t-1] - \rho_{ji}[2t-2])$, $w_i[2t-1] \leftarrow \sum_{j \in \mathcal{I}_i} (\tilde{\rho}_{ji}[2t-1] - \tilde{\rho}_{ji}[2t-2])$;
 - 4 $x_i[2t-1] \leftarrow \prod_{\mathcal{X}}^{\psi} \left(\frac{z_i[2t-1]}{w_i[2t-1]}, \alpha[2t-2] \right)$;
-

Due to the existence of self-loops, $w_i[t] \neq 0 \in \mathbb{R}$ for all $i \in \mathcal{V}$ and all $t \geq 0$. Thus, step 9 of Algorithm 1, and step 4 of Algorithm 2 are both well-defined. If an incoming link (j, i) is unreliable, since no new messages (“mass”) are received, ρ_{ji} and $\tilde{\rho}_{ji}$ are unchange, i.e., $\rho_{ji}[t] = \rho_{ji}[t-1]$ and $\tilde{\rho}_{ji}[t] = \tilde{\rho}_{ji}[t-1]$. On the other hand, when the incoming link (j, i) is reliable, we have $\rho_{ji}[t] = \sigma_j[t]$, and $\tilde{\rho}_{ji}[t] = \tilde{\sigma}_j[t]$. This explains steps 3-7 of Algorithm 1, as well as step 2 of Algorithm 2. In addition, it follows that, the total new weighted gradients and weights received by node i at iteration t is given by $\sum_{j \in \mathcal{I}_i} (\rho_{ji}[t] - \rho_{ji}[t-1])$ and $\sum_{j \in \mathcal{I}_i} (\tilde{\rho}_{ji}[t] - \tilde{\rho}_{ji}[t-1])$, respectively, which explains step 8 of Algorithm 1 and step 3 of Algorithm 2.

3.3 Augmented Graph

For a given $G(\mathcal{V}, \mathcal{E})$, augmented graph, denoted as $G^a(\mathcal{V}^a, \mathcal{E}^a)$, is constructed as follows [26]: (1) $\mathcal{V}^a = \mathcal{V} \cup \mathcal{E}$, i.e., $|\mathcal{E}|$ additional auxiliary nodes are introduced, each of which represents a link in $G(\mathcal{V}, \mathcal{E})$; (2) $\mathcal{E} \subseteq \mathcal{E}^a$, i.e., the edge set in $G^a(\mathcal{V}^a, \mathcal{E}^a)$ preserves the topology of $G(\mathcal{V}, \mathcal{E})$; (3) for

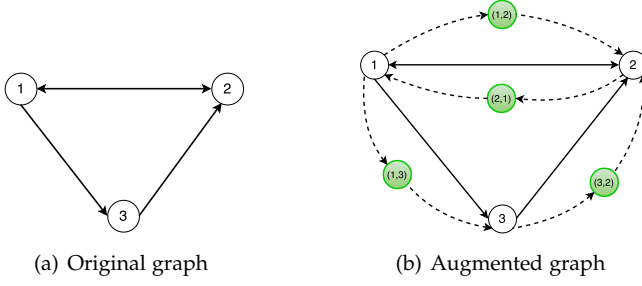


Fig. 1: For each directed link, a buffer node is added.

each $k \in \mathcal{V}$ and $(i, j) \in \mathcal{E}$, $(k, (i, j)) \in \mathcal{E}^a$ if and only if $k = i$, and $((i, j), k) \in \mathcal{E}^a$ if and only if $j = k$; (4) each auxiliary node has a self-loop, i.e., $((i, j), (i, j)) \in \mathcal{E}^a$. As shown in Fig. 1, where self-loops at individual nodes are not depicted, in the augmented graph (i.e., Fig. 1(b)), four additional nodes are introduced, each of which corresponds to a directed edge of the original graph. Note that each green node in Fig. 1(b) should also have a self-loop (which is not depicted).

Since $G(\mathcal{V}, \mathcal{E})$ is assumed to be strongly connected, its augmented graph is also strongly connected. For ease of future reference, we index the nodes in \mathcal{V} and the buffer nodes associated with \mathcal{E} consistently as follows. Define $m = n + |\mathcal{E}|$. Let $\phi : \mathcal{E} \rightarrow \{n + 1, \dots, m\}$ be an arbitrary bijection that maps $(i, j) \in \mathcal{E}$ to an integer within the range $\{n + 1, \dots, m\}$. For example, for Fig. 1(b), we can re-index the green nodes $(1, 2)$, $(2, 1)$, $(1, 3)$, $(3, 2)$ as 4, 5, 6, 7, respectively.

Recall that z_i and w_i are the weighted gradient and weight for $i \in \mathcal{V} = \{1, \dots, n\}$. For $k = \phi((j, i)) \in \{n + 1, \dots, m\}$, we define z_k and w_k as

$$z_k[t] \triangleq \sigma_j[t] - \rho_{ji}[t], \quad (8)$$

and

$$w_k[t] \triangleq \tilde{\sigma}_j[t] - \tilde{\rho}_{ji}[t], \quad (9)$$

with $z_k[0] = \mathbf{0} \in \mathbb{R}^d$ and $w_k[0] = 0 \in \mathbb{R}$. For each buffer node $i \in \{n + 1, \dots, m\}$, define $g_i[t] \triangleq \mathbf{0} \in \mathbb{R}^d$.

Next we constructively show that the evolution of $z_i[t], w_i[t]$ for $i \in \{1, \dots, m\}$ in Algorithm 1 and Algorithm 2 can be expressed in a matrix form. This is crucial in establishing the convergence of RPSDA.

3.4 Matrix Representation

For each link $(j, i) \in \mathcal{E}$, and $t \geq 1$, define the indicator variable $B_{(j,i)}[t]$ as follows:

$$B_{(j,i)}[t] \triangleq \begin{cases} 1, & \text{if link } (j, i) \text{ is reliable at time } t; \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

Note that messages are only transmitted among agents in **even** iterations, which is equivalent to the scenario that no links (excluding self-loops) are reliable at **odd** iterations. Thus, when t is **odd**, (10) becomes

$$B_{(j,i)}[t] \triangleq \begin{cases} 1, & \text{if } j = i; \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

We will now reformulate step 8 of Algorithm 1 and step 3 of Algorithm 2, and show that the evolution of z_i and w_i for $i \in \{1, \dots, m\}$ can be described in the following matrix form

$$z_i[t] = \sum_{j=1}^m z_j[t-1] \mathbf{M}_{ji}[t] + g_i[t-1], \quad (12)$$

and

$$w_i[t] = \sum_{j=1}^m w_j[t-1] \mathbf{M}_{ji}[t], \quad (13)$$

where the transition matrix $\mathbf{M}[t]$ (specified later) is a function of the indicator variables defined in (10) and (11).

From steps 3 – 7 of Algorithm 1, step 2 of Algorithm 2, (10), and (11), we get

$$\rho_{ji}[t] = B_{(j,i)}[t] \sigma_j[t] + (1 - B_{(j,i)}[t]) \rho_{ji}[t-1], \quad (14)$$

and

$$\tilde{\rho}_{ji}[t] = B_{(j,i)}[t] \tilde{\sigma}_j[t] + (1 - B_{(j,i)}[t]) \tilde{\rho}_{ji}[t-1]. \quad (15)$$

For $t \geq 1$ and $k = \phi((j, i))$, where $(j, i) \in \mathcal{E}$, from (9), (14) and (15), we have

$$z_k[t] = (1 - B_{(j,i)}[t]) \left(\frac{z_j[t-1]}{d_j^{\text{out}}} + z_k[t-1] \right), \quad (16)$$

and

$$w_k[t] = (1 - B_{(j,i)}[t]) \left(\frac{w_j[t-1]}{d_j^{\text{out}}} + w_k[t-1] \right). \quad (17)$$

Similarly, from step 8 of Algorithm 1, step 3 of Algorithm 2, (14), and (15), for each $i \in \mathcal{V}$, the updates of z_i and w_i are

$$z_i[t] = \sum_{j \in \mathcal{I}_i} B_{(j,i)}[t] \left(\frac{z_j[t-1]}{d_j^{\text{out}}} + z_{\phi((j,i))}[t-1] \right) + g_i[t], \quad (18)$$

and

$$w_i[t] = \sum_{j \in \mathcal{I}_i} B_{(j,i)}[t] \left(\frac{w_j[t-1]}{d_j^{\text{out}}} + w_{\phi((j,i))}[t-1] \right). \quad (19)$$

With the representation of updates of z_i and w_i in (16), (17), (18), and (19), we construct a matrix $\mathbf{M}[t] \in \mathbb{R}^{m \times m}$ with the following structure: For $i, j \in \mathcal{V} = \{1, \dots, n\}$ and $k' \in \{n + 1, \dots, m\}$ such that $(i, j) \in \mathcal{E}$ and $k' = \phi((i, j))$, let

$$\mathbf{M}_{ij}[t] \triangleq \frac{B_{(i,j)}[t]}{d_i^{\text{out}}}, \text{ and } \mathbf{M}_{ik'}[t] \triangleq \frac{1 - B_{(i,j)}[t]}{d_i^{\text{out}}}, \quad (20)$$

$$\mathbf{M}_{k'j}[t] \triangleq B_{(i,j)}[t], \text{ and } \mathbf{M}_{k'k'}[t] \triangleq 1 - B_{(i,j)}[t], \quad (21)$$

and any other entry in $\mathbf{M}[t]$ be 0.

Clearly, the obtained matrix $\mathbf{M}[t]$ is row stochastic for $t \geq 1$. It is easy to check that the matrix $\mathbf{M}[t]$ defined above satisfies (12) and (13). Let $\Psi(r, t)$ be the product of $t - r + 1$ row-stochastic matrices

$$\Psi(r, t) \triangleq \prod_{\tau=r}^t \mathbf{M}[\tau] = \mathbf{M}[r] \mathbf{M}[r+1] \cdots \mathbf{M}[t],$$

with $r \leq t$. In addition, $\Psi(t + 1, t) \triangleq \mathbf{I}$ by convention.

For $t \geq 1$, for each $i \in \{1, \dots, m\}$, expanding (12) out we have

$$z_i[t] = \sum_{r=0}^{t-1} \sum_{j=1}^m g_j[r] \Psi_{ji}(r+2, t). \quad (22)$$

Similar to (22), for the weight evolution, for each $i \in \{1, \dots, m\}$, we have

$$w_i[t] = \sum_{j=1}^m w_j[0] \Psi_{ji}(1, t) = \sum_{j=1}^n w_j[0] \Psi_{ji}(1, t), \quad (23)$$

where the last equality holds due to $w_i[0] = 1$ for $i \in \mathcal{V}$, and $w_i[0] = 0$ otherwise.

3.5 Convergence Properties of $\Psi(r, t)$

We investigate the convergence behavior of $\Psi(r, t)$ (where $r \leq t$) using ergodic coefficients and some celebrated results obtained by Hajnal [10].

Given a row stochastic matrix \mathbf{A} , coefficients of ergodicity $\delta(\mathbf{A})$ and $\lambda(\mathbf{A})$ are defined as:

$$\delta(\mathbf{A}) \triangleq \max_j \max_{i_1, i_2} |\mathbf{A}_{i_1 j} - \mathbf{A}_{i_2 j}|, \quad (24)$$

and

$$\lambda(\mathbf{A}) \triangleq 1 - \min_{i_1, i_2} \sum_j \min\{\mathbf{A}_{i_1 j}, \mathbf{A}_{i_2 j}\}. \quad (25)$$

Informally speaking, the coefficients of ergodicity defined in (24) and (25) characterize the “difference” between any pair of rows of the given row-stochastic matrix \mathbf{A} . It is easy to see that $0 \leq \delta(\mathbf{A}) \leq 1$, $0 \leq \lambda(\mathbf{A}) \leq 1$, and that the rows of \mathbf{A} are identical if and only if $\delta(\mathbf{A}) = 0 = \lambda(\mathbf{A})$. In addition, the ergodic coefficients $\delta(\cdot)$ and $\lambda(\cdot)$ have the following connection.

Proposition 1. [10] For any p square row stochastic matrices $\mathbf{Q}[1], \mathbf{Q}[2], \dots, \mathbf{Q}[p]$, it holds that

$$\delta(\mathbf{Q}[1]\mathbf{Q}[2] \dots \mathbf{Q}[p]) \leq \prod_{k=1}^p \lambda(\mathbf{Q}[k]). \quad (26)$$

Proposition 1 implies that if $\lambda(\mathbf{Q}[k]) \leq 1 - c$ for some $c > 0$ and for all $1 \leq k \leq p$, then $\delta(\mathbf{Q}[1], \mathbf{Q}[2] \dots \mathbf{Q}[p])$ goes to zero exponentially fast as p increases. Next we show that, for sufficiently large $k - r$, it holds that

$$\lambda(\Psi(r, k)) \leq 1 - \beta^{2nB},$$

where

$$\beta \triangleq \frac{1}{\max_{i \in \mathcal{V}} (d_i^{\text{out}})}. \quad (27)$$

To prove this claim, we need the following lemma.

Lemma 1. Suppose that $2t + 2 - r \geq 2nB$ and $B \geq 1$. Then every entry in $\Psi(r, 2t + 1)$ is lower bounded by β^{2nB} , i.e., for $i, j \in \{1, \dots, n, n+1, \dots, m\}$, it holds that

$$\Psi_{ij}(r, 2t + 1) \geq \beta^{2nB}.$$

Proof. Since

$$\begin{aligned} \Psi(r, 2t + 1) &= \Psi(r, 2t + 1 - 2nB) \\ &\quad \times \Psi(2t + 2 - 2nB, 2t + 1), \end{aligned}$$

and $\Psi(r, 2t + 1 - 2nB)$ is row-stochastic, to prove Lemma 1, it is enough to show that every entry in

$$\Psi(2t + 2 - 2nB, 2t + 1)$$

is lower bounded by β^{2nB} .

Note that $\Psi(2t + 2 - 2nB, 2t + 1)$ is a product of $2nB$ matrices defined in (20) and (21). Since the underlying communication network $G(\mathcal{V}, \mathcal{E})$ is assumed to be strongly connected, for any $i, j \in \mathcal{V} = \{1, \dots, n\}$, there exists an i, j -path of length at most n in $G(\mathcal{V}, \mathcal{E})$. In addition, since every node contained in the i, j -path has a self-loop, and every edge contained is reliable at least once in every B message transmissions (i.e., in every $2B$ iterations), the following holds

$$\Psi_{ij}(2t + 2 - 2nB, 2t + 1) \geq \beta^{2nB}, \quad \forall i, j \in \mathcal{V}.$$

For any $k \in \{n+1, \dots, m\}$, since $\phi(\cdot)$ is a bijection from \mathcal{E} to $\{n+1, \dots, m\}$, there exists an edge $(p, q) \in \mathcal{E}$ such that $\phi^{-1}(k) = (p, q)$. We have

$$\begin{aligned} \Psi_{ki}(2t + 2 - 2nB, 2t + 1) &= \sum_{k'=1}^m \left(\Psi_{kk'}(2t + 2 - 2nB, 2t + 1 - 2(n-1)B) \right. \\ &\quad \times \Psi_{k'i}(2t + 2 - 2(n-1)B, 2t + 1) \Big) \\ &\geq \Psi_{kq}(2t + 2 - 2nB, 2t + 1 - 2(n-1)B) \\ &\quad \times \Psi_{qi}(2t + 2 - 2(n-1)B, 2t + 1) \\ &\geq \Psi_{kq}(2t + 2 - 2nB, 2t + 1 - 2(n-1)B) \times \beta^{2(n-1)B} \\ &\geq \beta^{2nB}, \end{aligned}$$

where the last inequality holds from the fact that link (p, q) is reliable at least once in every $2B$ iterations.

By the matrix construction in (20), we have

$$\mathbf{M}_{pk}[2t + 1] = \frac{1}{d_p^{\text{out}}} \geq \beta,$$

where $k = \phi((p, q))$ and $(p, q) \in \mathcal{E}$. We get

$$\begin{aligned} \Psi_{ik}(2t + 2 - 2nB, 2t + 1) &\geq \Psi_{ip}(2t + 2 - 2nB, 2t) \mathbf{M}_{pk}[2t + 1] \\ &\geq \beta^{2nB}. \end{aligned} \quad (28)$$

Similarly, we can show that for any $k_1, k_2 \in \{n+1, \dots, m\}$ such that $\phi^{-1}(k_1) = (p_1, q_1) \in \mathcal{E}$ and $\phi^{-1}(k_2) = (p_2, q_2) \in \mathcal{E}$, the following holds.

$$\Psi_{k_1 k_2}(2t + 2 - 2nB, 2t + 1) \geq \beta^{2nB}.$$

The proof of Lemma 1 is complete. \square

By Proposition 1 and Lemma 1, we can show Lemma 2, which says that the difference between any pair of rows in $\Psi(r, t)$ diminishes exponentially fast.

Lemma 2. For $r \leq t$, it holds that

$$\delta(\Psi(r, t)) \leq \gamma^{\lfloor \frac{t-r}{2nB} \rfloor},$$

where $\gamma = 1 - \beta^{2nB}$.

Proof. Define

$$\mathbf{Q}[k] \triangleq \Psi(r + 2(k-1)nB, r + 2knB - 1). \quad (29)$$

When r is **even**, from Lemma 1, and definition of $\lambda(\cdot)$ in (25), we have

$$\lambda(\mathbf{Q}[k]) \leq 1 - \beta^{2nB} = \gamma. \quad (30)$$

When $r \leq t$, $\Psi(r, t)$ can be rewritten as follows.

$$\begin{aligned} \Psi(r, t) &= \Psi\left(r, r + \lfloor \frac{t-r+1}{2nB} \rfloor \cdot 2nB - 1\right) \\ &\quad \times \Psi\left(r + \lfloor \frac{t-r+1}{2nB} \rfloor \cdot 2nB, t\right) \\ &= \mathbf{Q}[1] \times \cdots \times \mathbf{Q}\left[\lfloor \frac{t-r+1}{2nB} \rfloor\right] \\ &\quad \times \Psi\left(r + \lfloor \frac{t-r+1}{2nB} \rfloor \cdot 2nB, t\right) \\ &= \left(\prod_{k=1}^{\lfloor \frac{t-r+1}{2nB} \rfloor} \mathbf{Q}[k]\right) \Psi\left(r + \lfloor \frac{t-r+1}{2nB} \rfloor \cdot 2nB, t\right). \end{aligned} \quad (31)$$

By Proposition 1 and (31), we have

$$\begin{aligned} \delta(\Psi(r, t)) &\leq \prod_{k=1}^{\lfloor \frac{t-r+1}{2nB} \rfloor} \lambda(\mathbf{Q}[k]) \lambda\left(\Psi\left(r + \lfloor \frac{t-r+1}{2nB} \rfloor \cdot 2nB, t\right)\right) \\ &\stackrel{(a)}{\leq} \prod_{k=1}^{\lfloor \frac{t-r+1}{2nB} \rfloor} \lambda(\mathbf{Q}[k]) \\ &\stackrel{(b)}{\leq} \gamma^{\lfloor \frac{t-r+1}{2nB} \rfloor} \\ &\leq \gamma^{\lfloor \frac{t-r}{2nB} \rfloor}, \end{aligned} \quad (32)$$

where inequality (a) follows from the fact that

$$\lambda\left(\Psi\left(r + \lfloor \frac{t-r+1}{2nB} \rfloor \cdot 2nB, t\right)\right) \leq 1,$$

and inequality (b) follows from (30). Note that (32) holds for any **even** r .

When r is **odd**, we consider the product $\Psi(r+1, t)$, and use the fact that

$$\delta(\Psi(r, t)) \leq \delta(\Psi(r+1, t)).$$

In particular,

$$\Psi(r, t) = \mathbf{M}[r] \Psi(r+1, t),$$

and

$$\begin{aligned} \delta(\Psi(r, t)) &= \delta(\mathbf{M}[r] \Psi(r+1, t)) \\ &\leq \max_j \max_{i_1, i_2} |\Psi_{i_1 j}(r+1, t) - \Psi_{i_1 j}(r, t)| \\ &= \delta(\Psi(r+1, t)). \end{aligned}$$

Since r is **odd**, then $r+1$ is **even**. By (32), we get

$$\delta(\Psi(r+1, t)) \leq \gamma^{\lfloor \frac{t-(r+1)+1}{2nB} \rfloor} = \gamma^{\lfloor \frac{t-r}{2nB} \rfloor}. \quad (33)$$

Therefore, the proof of Lemma 2 is complete. \square

4 CONVERGENCE OF RPSDA

In this section, we analyze the convergence of RPSDA. Our proofs parallel the structure of proofs in [24], but with some key differences to take into account the link failures.

Let $\bar{z}[t] \triangleq \frac{1}{n} \sum_{i=1}^m z_i[t]$ be the weighted average of $z_i[t]$ over all nodes in the augmented graph. We have

$$\begin{aligned} \bar{z}[t] &= \frac{1}{n} \sum_{i=1}^m z_i[t] = \frac{1}{n} \sum_{r=0}^{t-1} \sum_{i=1}^m g_i[r] \\ &= \frac{1}{n} \sum_{r=0}^{t-1} \sum_{i=1}^n g_i[r], \end{aligned} \quad (34)$$

where the last equality follows from the fact that $g_i[r] = 0$ for each $i \in \{n+1, \dots, m\}$ and $r \geq 0$. We also need the sequence $\{y(t)\}_{t=1}^\infty$ that is defined by the projection of $\bar{z}[t]$:

$$y[t] \triangleq \prod_{\mathcal{X}}^{\psi}(\bar{z}[t], \alpha[t-1]). \quad (35)$$

We define the running averages of $x_i[t]$ and $y[t]$ at **odd** iterations, denoted by $\hat{x}_i[T]$ and $\hat{y}_i[T]$, respectively, as follows:

$$\hat{x}_i[T] = \frac{1}{T} \sum_{t=1}^T x_i[2t-1], \quad \text{and} \quad \hat{y}_i[T] = \frac{1}{T} \sum_{t=1}^T y[2t-1].$$

Following the same line of analysis in [24], we are able to show Lemma 3 (stated below).

Lemma 3. *For any $x^* \in \mathcal{X}$, it holds that*

$$\begin{aligned} h(\hat{x}_j[T]) - h(x^*) &\leq \frac{L^2}{2T} \sum_{t=1}^T \alpha[2t-2] + \frac{1}{T\alpha[2T]} \psi(x^*) \\ &\quad + \frac{L}{T} \sum_{t=1}^T \alpha[2t-2] \left\| \bar{z}[2t-1] - \frac{z_j[2t-1]}{w_j[2t-1]} \right\| \\ &\quad + \frac{2L}{nT} \sum_{t=1}^T \sum_{i=1}^n \alpha[2t-2] \left\| \bar{z}[2t-1] - \frac{z_i[2t-1]}{w_i[2t-1]} \right\|. \end{aligned}$$

Note that the summation on the right hand side is taken over all nodes in the *original graph* $G(\mathcal{V}, \mathcal{E})$ rather than the augmented graph $G^a(\mathcal{V}^a, \mathcal{E}^a)$. Lemma 3 is proved in Appendix A.

To complete the convergence analysis, we need to bound $\left\| \bar{z}[2t-1] - \frac{z_i[2t-1]}{w_i[2t-1]} \right\|$ for any node i and any iteration $t \geq 1$. Our analysis is different from that in [24] due to $\mathbf{M}[t]$'s dependency on t .

Lemma 4. *When $t \geq nB + 1$, for each $i \in \mathcal{V}$, it holds that*

$$\left\| \bar{z}[2t-1] - \frac{z_i[2t-1]}{w_i[2t-1]} \right\| \leq \frac{L}{\beta^{2nB} \gamma^{\frac{2nB+1}{2nB}} (1 - \gamma^{\frac{1}{2nB}})}.$$

Proof. Recall that $w_i[0] = 1$, for all $i \in \mathcal{V}$, then by (22), (23) and (34) we obtain

$$\begin{aligned} & \left\| \bar{z}[2t-1] - \frac{z_i[2t-1]}{w_i[2t-1]} \right\| \\ &= \left\| \sum_{r=0}^{2t-2} \sum_{j=1}^n \left(\frac{1}{n} g_j[r] - \frac{\Psi_{ji}(r+2, 2t-1) g_j[r]}{\sum_{s=1}^n \Psi_{si}(1, 2t-1)} \right) \right\| \\ &\leq \sum_{r=0}^{2t-2} \sum_{j=1}^n \|g_j[r]\| \left| \frac{1}{n} - \frac{\Psi_{ji}(r+2, 2t-1)}{\sum_{s=1}^n \Psi_{si}(1, 2t-1)} \right| \\ &\leq L \sum_{r=0}^{2t-2} \sum_{j=1}^n \left| \frac{1}{n} - \frac{\Psi_{ji}(r+2, 2t-1)}{\sum_{s=1}^n \Psi_{si}(1, 2t-1)} \right|, \end{aligned} \quad (36)$$

where the last inequality is true since $\|g_j[r]\| \leq L$.

We now show that the right hand side of (36) is bounded. Specifically,

$$\begin{aligned} & \left| \frac{1}{n} - \frac{\Psi_{ji}(r+2, 2t-1)}{\sum_{s=1}^n \Psi_{si}(1, 2t-1)} \right| \\ &= \left| \frac{\sum_{s=1}^n \Psi_{si}(1, 2t-1) - n \Psi_{ji}(r+2, 2t-1)}{n \sum_{s=1}^n \Psi_{si}(1, 2t-1)} \right| \\ &= \left| \frac{\sum_{s=1}^n (\Psi_{si}(1, 2t-1) - \Psi_{ji}(r+2, 2t-1))}{n \sum_{s=1}^n \Psi_{si}(1, 2t-1)} \right| \\ &\leq \frac{\sum_{s=1}^n |\Psi_{si}(1, 2t-1) - \Psi_{ji}(r+2, 2t-1)|}{|n \sum_{s=1}^n \Psi_{si}(1, 2t-1)|}. \end{aligned} \quad (37)$$

In fact, when $t \geq nB+1$, the right hand side of (37) goes to 0 exponentially fast, shown as follows. By Lemma 1, when $2t-1-1 \geq 2nB$, we get

$$\begin{aligned} & \frac{\sum_{s=1}^n |\Psi_{si}(1, 2t-1) - \Psi_{ji}(r+2, 2t-1)|}{|n \sum_{s=1}^n \Psi_{si}(1, 2t-1)|} \\ &\leq \frac{\sum_{s=1}^n |\Psi_{si}(1, 2t-1) - \Psi_{ji}(r+2, 2t-1)|}{n^2 \beta^{2nB}}. \end{aligned} \quad (38)$$

Since

$$\Psi_{si}(1, 2t-1) = \sum_{k=1}^m \Psi_{sk}(1, r+1) \Psi_{ki}(r+2, 2t-1),$$

and $\sum_{k=1}^m \Psi_{sk}(1, r+1) = 1$, it holds that

$$\begin{aligned} & |\Psi_{si}(1, 2t-1) - \Psi_{ji}(r+2, 2t-1)| \\ &\leq \sum_{k=1}^m \Psi_{sk}(1, r+1) |\Psi_{ki}(r+2, 2t-1) - \Psi_{ji}(r+2, 2t-1)| \\ &\leq \sum_{k=1}^m \Psi_{sk}(1, r+1) \gamma^{\lfloor \frac{2t-r-3}{2nB} \rfloor} \quad \text{by Lemma 2} \\ &= \gamma^{\lfloor \frac{2t-r-3}{2nB} \rfloor}, \end{aligned} \quad (39)$$

where the last equality holds since $\sum_{k=1}^m \Psi_{sk}(1, r+1) = 1$.

Plugging (39) back into (38), we get

$$\begin{aligned} & \frac{\sum_{s=1}^n |\Psi_{si}(1, 2t-1) - \Psi_{ji}(r+2, 2t-1)|}{|n \sum_{s=1}^n \Psi_{si}(1, 2t-1)|} \\ &\leq \frac{\sum_{s=1}^n \gamma^{\lfloor \frac{2t-r-3}{2nB} \rfloor}}{n^2 \beta^{2nB}} = \frac{\gamma^{\lfloor \frac{2t-r-3}{2nB} \rfloor}}{n \beta^{2nB}}. \end{aligned} \quad (40)$$

By (36), (37), and (40), we obtain

$$\begin{aligned} \left\| \bar{z}[2t-1] - \frac{z_i[2t-1]}{w_i[2t-1]} \right\| &\leq L \sum_{r=0}^{2t-2} \sum_{j=1}^n \frac{\gamma^{\lfloor \frac{2t-r-3}{2nB} \rfloor}}{n \beta^{2nB}} \\ &\leq L \sum_{r=0}^{2t-2} \frac{\gamma^{\frac{2t-r-3}{2nB} - 1}}{\beta^{2nB}} \\ &\leq \frac{L}{\beta^{2nB} \gamma^{\frac{2nB+1}{2nB}} (1 - \gamma^{\frac{1}{2nB}})}, \end{aligned}$$

where we used the formula for a finite geometric sum. Therefore, the proof of Lemma 4 is complete. \square

Theorem 1. Let $x^* \in X^*$, and suppose that $\psi(x^*) \leq R^2$. Let $\{\alpha[t] = \frac{A}{\sqrt{t}}\}_{t=1}^\infty$ with $\alpha[0] = A$ be the sequence of stepsizes used in step 9 of Algorithm 1 and step 4 of Algorithm 2 for some positive constant A . Then, for $T \geq nB+1$, we have for all $j \in \mathcal{V}$

$$\begin{aligned} h(\hat{x}_j[T]) - h(x^*) &\leq \frac{\sqrt{2}L^2 A}{\sqrt{T}} + \frac{\sqrt{2}R^2}{A\sqrt{T}} \\ &\quad + \frac{6\sqrt{2}L^2 A}{\beta^{2nB} \gamma^{\frac{2nB+1}{2nB}} (1 - \gamma^{\frac{1}{2nB}}) \sqrt{T}}. \end{aligned}$$

Proof. From Lemma 3, we have

$$\begin{aligned} h(\hat{x}_j[T]) - h(x^*) &\leq \frac{L^2}{2T} \sum_{t=1}^T \alpha[2t-2] + \frac{1}{T\alpha[2T]} \psi(x^*) \\ &\quad + \frac{2L}{nT} \sum_{t=1}^T \sum_{i=1}^n \alpha[2t-2] \left\| \bar{z}[2t-1] - \frac{z_i[2t-1]}{w_i[2t-1]} \right\| \\ &\quad + \frac{L}{T} \sum_{t=1}^T \alpha[2t-2] \left\| \bar{z}[2t-1] - \frac{z_j[2t-1]}{w_j[2t-1]} \right\| \\ &\stackrel{(a)}{\leq} \frac{L^2}{2T} \sum_{t=1}^T \alpha[2t-2] + \frac{1}{T\alpha[2T]} R^2 \\ &\quad + \frac{2L}{nT} \sum_{t=1}^T \sum_{i=1}^n \alpha[2t-2] \left\| \bar{z}[2t-1] - \frac{z_i[2t-1]}{w_i[2t-1]} \right\| \\ &\quad + \frac{L}{T} \sum_{t=1}^T \alpha[2t-2] \left\| \bar{z}[2t-1] - \frac{z_j[2t-1]}{w_j[2t-1]} \right\| \\ &\stackrel{(b)}{\leq} \frac{L^2}{2T} \sum_{t=1}^T \alpha[2t-2] + \frac{1}{T\alpha[2T]} R^2 \\ &\quad + \frac{2L}{nT} \sum_{t=1}^T \sum_{i=1}^n \alpha[2t-2] \frac{L}{\beta^{2nB} \gamma^{\frac{2nB+1}{2nB}} (1 - \gamma^{\frac{1}{2nB}})} \\ &\quad + \frac{L}{T} \sum_{t=1}^T \alpha[2t-2] \frac{L}{\beta^{2nB} \gamma^{\frac{2nB+1}{2nB}} (1 - \gamma^{\frac{1}{2nB}})} \\ &= \frac{L^2}{2T} \sum_{t=1}^T \alpha[2t-2] + \frac{R^2}{T\alpha[2T]} \\ &\quad + \frac{3L^2}{\beta^{2nB} \gamma^{\frac{2nB+1}{2nB}} (1 - \gamma^{\frac{1}{2nB}}) T} \sum_{t=1}^T \alpha[2t-2]. \end{aligned} \quad (41)$$

where inequality (a) is true sine $\psi(x^*) \leq R^2$, and inequality (b) follows from Lemma 4.

Since $\alpha[t] = \frac{A}{\sqrt{t}}$ with $\alpha[0] = A$, we have

$$\sum_{t=1}^T \alpha[2t-2] \leq \sum_{t=1}^{2T} \alpha[t] = \sum_{t=1}^{2T} \frac{A}{\sqrt{t}} \leq 2\sqrt{2T}A. \quad (42)$$

In addition,

$$\frac{1}{T\alpha[2T]} = \frac{1}{T\frac{A}{\sqrt{2T}}} \leq \frac{\sqrt{2T}}{AT} \leq \frac{\sqrt{2}}{A\sqrt{T}}. \quad (43)$$

With (42) and (43), the right hand side of (41) can be bounded as

$$\begin{aligned} h(\hat{x}_j[T]) - h(x^*) &\leq \frac{L^2}{2T} \sum_{t=1}^T \alpha[2t-2] + \frac{1}{T\alpha[2T]} R^2 \\ &\quad + \frac{3L^2}{\beta^{2nB} \gamma^{\frac{2nB+1}{2nB}} \left(1 - \gamma^{\frac{1}{2nB}}\right) T} \sum_{t=1}^T \alpha[2t-2] \\ &\leq \frac{L^2}{2T} 2\sqrt{2T}A + \frac{\sqrt{2}}{A\sqrt{T}} R^2 \\ &\quad + \frac{3L^2}{\beta^{2nB} \gamma^{\frac{2nB+1}{2nB}} \left(1 - \gamma^{\frac{1}{2nB}}\right) T} 2\sqrt{2T}A \\ &= \frac{\sqrt{2}L^2A}{\sqrt{T}} + \frac{\sqrt{2}R^2}{A\sqrt{T}} + \frac{6\sqrt{2}L^2A}{\beta^{2nB} \gamma^{\frac{2nB+1}{2nB}} \left(1 - \gamma^{\frac{1}{2nB}}\right) \sqrt{T}}, \end{aligned}$$

proving the theorem. \square

Therefore, the algorithm will converge, and the convergence rate is $O(\frac{1}{\sqrt{T}})$.

Note that Theorem 1 holds for any positive constant A . Optimizing over A , the constant hidden in $O(\frac{1}{\sqrt{T}})$ can be improved. In particular, the optimal value of A has the following expression

$$A^* = \frac{R}{L} \left(1 + \frac{6}{\beta^{2nB} \gamma^{\frac{2nB+1}{2nB}} \left(1 - \gamma^{\frac{1}{2nB}}\right)} \right)^{-\frac{1}{2}}.$$

5 CONCLUSION

We study the multi-agent optimization problem in the presence of link failures. We propose a distributed optimization algorithm that is robust to packet-dropping link failures by combining the Push-Sum Distributed Dual Averaging algorithm [24] with a robust average consensus algorithm [26]. In our algorithm each agent updates its local estimate using slightly different routines in odd and even iterations. We show that these local estimates converge to a common optimum of $h(\cdot)$ sub-linearly at convergence rate $O(1/\sqrt{t})$. The main analysis challenges come from the fact that the effective communication graph is time varying, and that each agent does not know the actual number of reliable outgoing links at each iteration.

APPENDIX A

We begin with restating two relevant lemmas proved in [7], [18], respectively.

Lemma 5. [18] For an arbitrary pair $u, v \in \mathbb{R}^d$, we have

$$\left\| \prod_{\mathcal{X}}^{\psi}(u, \alpha) - \prod_{\mathcal{X}}^{\psi}(v, \alpha) \right\| \leq \alpha \|u - v\|.$$

Let $\{g[t]\}_{t=0}^{\infty} \subseteq \mathbb{R}^d$ be an arbitrary sequence, and consider the sequence $\{x[t]\}_{t=1}^{\infty}$ generated as follows

$$x[t+1] \triangleq \prod_{\mathcal{X}}^{\psi} \left(\sum_{r=0}^t g[r], \alpha[t] \right). \quad (44)$$

Lemma 6. [7] For any non-increasing sequence of stepsize $\{\alpha[t]\}_{t=0}^{\infty}$, and for any $x^* \in \mathcal{X}$

$$\sum_{t=1}^T \langle g[t], x[t] - x^* \rangle \leq \frac{1}{2} \sum_{t=1}^T \alpha[t-1] \|g[t]\|^2 + \frac{1}{\alpha[T]} \psi(x^*). \quad (45)$$

PROOF OF LEMMA 3

Proof. For $T \geq 1$, we have

$$\begin{aligned} h(\hat{x}_j[T]) - h(x^*) &= h(\hat{y}[T]) - h(x^*) + h(\hat{x}_j[T]) - h(\hat{y}[T]) \\ &\leq h(\hat{y}[T]) - h(x^*) + L \|\hat{x}_j[T] - \hat{y}[T]\| \\ &\leq \frac{1}{T} \sum_{t=1}^T (h(y[2t-1]) - h(x^*)) \\ &\quad + \frac{L}{T} \sum_{t=1}^T \|x_j[2t-1] - y[2t-1]\| \\ &= \frac{1}{T} \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n (h_i(y[2t-1]) - h_i(x^*)) \\ &\quad + \frac{L}{T} \sum_{t=1}^T \|x_j[2t-1] - y[2t-1]\|. \end{aligned} \quad (46)$$

The first inequality holds from L -Lipschitz continuity; and the second inequality is true due to the convexity of h as well as the definition of the running averages $\hat{x}_j[T]$ and $\hat{y}[T]$. Now adding and subtracting

$\frac{1}{T} \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n h_i(x_i[2t-1])$ from (46), we get

$$\begin{aligned}
& h(\hat{x}_j[T]) - h(x^*) \\
& \leq \frac{1}{T} \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n (h_i(y[2t-1]) - h_i(x_i[2t-1])) \\
& \quad + \frac{1}{T} \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n (h_i(x_i[2t-1]) - h_i(x^*)) \\
& \quad + \frac{L}{T} \sum_{t=1}^T \|x_j[2t-1] - y[2t-1]\| \\
& \stackrel{(a)}{\leq} \frac{1}{T} \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n L \|x_i[2t-1] - y[2t-1]\| \\
& \quad + \frac{1}{T} \sum_{t=1}^T \frac{1}{n} \sum_{i=1}^n \langle g_i[2t-1], x_i[2t-1] - x^* \rangle \\
& \quad + \frac{L}{T} \sum_{t=1}^T \|x_j[2t-1] - y[2t-1]\| \\
& \stackrel{(b)}{\leq} \frac{L}{Tn} \sum_{t=1}^T \alpha[2t-2] \sum_{i=1}^n \left\| \frac{z_i[2t-1]}{w_i[2t-1]} - \bar{z}[2t-1] \right\| \\
& \quad + \frac{1}{Tn} \sum_{t=1}^T \sum_{i=1}^n \langle g_i[2t-1], x_i[2t-1] - x^* \rangle \\
& \quad + \frac{L}{T} \sum_{t=1}^T \alpha[2t-2] \left\| \frac{z_j[2t-1]}{w_j[2t-1]} - \bar{z}[2t-1] \right\|, \quad (47)
\end{aligned}$$

where inequality (a) follows because for each $i \in \mathcal{V}$, $h_i(\cdot)$ is L -Lipschitz continuous and convex; and inequality (b) holds due to the update of y in (35), the update of x_i in Algorithms 1 and 2, and Lemma 5.

For the second term in (47), we have

$$\begin{aligned}
& \sum_{i=1}^n \langle g_i[2t-1], x_i[2t-1] - x^* \rangle \\
& = \sum_{i=1}^n \langle g_i[2t-1], y[2t-1] - x^* \rangle \\
& \quad + \sum_{i=1}^n \langle g_i[2t-1], x_i[2t-1] - y[2t-1] \rangle \\
& = \left\langle \sum_{i=1}^n g_i[2t-1], y[2t-1] - x^* \right\rangle \\
& \quad + \sum_{i=1}^n \langle g_i[2t-1], x_i[2t-1] - y[2t-1] \rangle. \quad (48)
\end{aligned}$$

Let

$$\tilde{g}[t] \triangleq \frac{1}{n} \sum_{i=1}^n g_i[2t-1]. \quad (49)$$

Since $g_i[2t] = 0$ for each $i \in \mathcal{V}$ and $t \geq 0$, it holds that

$$\begin{aligned}
\bar{z}[2t-1] &= \frac{1}{n} \sum_{r=0}^{2t-2} \sum_{i=1}^n g_i[r] \quad \text{by (34)} \\
&= \frac{1}{n} \sum_{r=1}^{t-1} \sum_{i=1}^n g_i[2r-1] \\
&= \sum_{r=1}^{t-1} \tilde{g}[r]. \quad (50)
\end{aligned}$$

Define

$$\tilde{\alpha}[t] \triangleq \alpha[2t], \quad \forall t \geq 0. \quad (51)$$

Since $\alpha[t]$ is non-increasing, $\tilde{\alpha}[t]$ is also non-increasing. Let $\tilde{y}[t] \triangleq y[2t-1]$, so we get

$$\begin{aligned}
\tilde{y}[t] &\triangleq y[2t-1] \\
&= \prod_{\mathcal{X}}^{\psi} (\bar{z}[2t-1], \alpha[2t-2]) \\
&= \prod_{\mathcal{X}}^{\psi} \left(\sum_{r=1}^{t-1} \tilde{g}[r], \tilde{\alpha}[t-1] \right). \quad \text{by (50) and (51)}
\end{aligned} \quad (52)$$

Thus, for the first term in the right hand side of (48), we get

$$\begin{aligned}
& \sum_{t=1}^T \frac{1}{n} \left\langle \sum_{i=1}^n g_i[2t-1], y[2t-1] - x^* \right\rangle \\
& = \sum_{t=1}^T \left\langle \frac{1}{n} \sum_{i=1}^n g_i[2t-1], y[2t-1] - x^* \right\rangle \\
& = \sum_{t=1}^T \langle \tilde{g}[t], \tilde{y}[t] - x^* \rangle \quad \text{by (49) and (52)} \\
& \stackrel{(a)}{\leq} \frac{1}{2} \sum_{t=1}^T \tilde{\alpha}[t-1] \|\tilde{g}[t]\|^2 + \frac{1}{\tilde{\alpha}[T]} \psi(x^*) \\
& \leq \frac{L^2}{2} \sum_{t=1}^T \alpha[2t-2] + \frac{1}{\alpha[2T]} \psi(x^*), \quad (53)
\end{aligned}$$

where inequality (a) follows from Lemma 6. In addition, for the second term in the right hand side of (48), we have

$$\begin{aligned}
& \sum_{i=1}^n \langle g_i[2t-1], x_i[2t-1] - y[2t-1] \rangle \\
& \leq \sum_{i=1}^n \|g_i[2t-1]\| \|x_i[2t-1] - y[2t-1]\| \\
& \leq L \sum_{i=1}^n \alpha[2t-2] \left\| \frac{z_i[2t-1]}{w_i[2t-1]} - \bar{z}[2t-1] \right\|, \quad (54)
\end{aligned}$$

where the last inequality follows from Lemma 5.

By (53), (54), and (47), we get

$$\begin{aligned}
h(\hat{x}_j[T]) - h(x^*) &\leq \frac{L^2}{2T} \sum_{t=1}^T \alpha[2t-2] + \frac{1}{T\alpha[2T]} \psi(x^*) \\
&\quad + \frac{2L}{nT} \sum_{t=1}^T \sum_{i=1}^n \alpha[2t-2] \left\| \bar{z}[2t-1] - \frac{z_i[2t-1]}{w_i[2t-1]} \right\| \\
&\quad + \frac{L}{T} \sum_{t=1}^T \alpha[2t-2] \left\| \bar{z}[2t-1] - \frac{z_j[2t-1]}{w_j[2t-1]} \right\|,
\end{aligned}$$

proving the lemma. \square

ACKNOWLEDGMENTS

This research is supported in part by National Science Foundation awards NSF 1329681 and 1421918. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies or the U.S. government.

REFERENCES

- [1] D. P. Bertsekas. *Convex Optimization Algorithms*. Athena Scientific, 2015.
- [2] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: numerical methods*. Prentice-Hall, Inc., 1989.
- [3] F. Bnizit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli. Weighted gossip: Distributed averaging using non-doubly stochastic matrices. In *Proceedings of IEEE International Symposium on Information Theory Proceedings (ISIT)*, pages 1753–1757, June 2010.
- [4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE/ACM Transactions on Networking (TON)*, 14(SI):2508–2530, 2006.
- [5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.
- [6] Y. Chen, R. Tron, A. Terzis, and R. Vidal. Corrective consensus: Converging to the exact average. In *Proceedings of IEEE Conference on Decision and Control (CDC)*, pages 1221–1228, December 2010.
- [7] J. Duchi, A. Agarwal, and M. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, 2012.
- [8] I. Eyal, I. Keidar, and R. Rom. Limosense: live monitoring in dynamic sensor networks. *Distributed Computing*, 27(5):313–328, 2014.
- [9] F. Fagnani and S. Zampieri. Average consensus with packet drop communication. *SIAM Journal on Control and Optimization*, 48(1):102–133, 2009.
- [10] J. Hajnal and M. Bartlett. Weak ergodicity in non-homogeneous markov chains. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 54, pages 233–246. Cambridge Univ Press, 1958.
- [11] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Automatic Control, IEEE Transactions on*, 48(6):988–1001, 2003.
- [12] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proceedings of IEEE Symposium on Foundations of Computer Science*, pages 482–491. IEEE, October 2003.
- [13] I. Lobel and A. Ozdaglar. Distributed subgradient methods for convex optimization over random networks. *Automatic Control, IEEE Transactions on*, 56(6):1291–1306, 2011.
- [14] A. Nedic and A. Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, 2015.
- [15] A. Nedić, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis. On distributed averaging algorithms and quantization effects. *Automatic Control, IEEE Transactions on*, 54(11):2506–2517, 2009.
- [16] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.
- [17] A. Nedić, A. Ozdaglar, and P. A. Parrilo. Constrained consensus and optimization in multi-agent networks. *Automatic Control, IEEE Transactions on*, 55(4):922–938, 2010.
- [18] Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009.
- [19] S. Patterson, B. Bamieh, and A. El Abbadi. Distributed average consensus with stochastic communication failures. In *Proceedings of IEEE Conference on Decision and Control (CDC)*, pages 4215–4220, December 2007.
- [20] S. S. Ram, A. Nedić, and V. V. Veeravalli. Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of optimization theory and applications*, 147(3):516–545, 2010.
- [21] L. Su and N. Vaidya. Byzantine multi-agent optimization: Part I. *arXiv preprint arXiv:1506.04681*, 2015.
- [22] A. Tahbaz-Salehi and A. Jadbabaie. A necessary and sufficient condition for consensus over random networks. *Automatic Control, IEEE Transactions on*, 53(3):791–795, 2008.
- [23] H. G. Tanner, A. Jadbabaie, and G. J. Pappas. Flocking in fixed and switching networks. *Automatic Control, IEEE Transactions on*, 52(5):863–868, 2007.
- [24] K. I. Tsianos, S. Lawlor, and M. G. Rabbat. Push-sum distributed dual averaging for convex optimization. In *Proceedings of IEEE Conference on Decision and Control (CDC)*, pages 5453–5458, December 2012.
- [25] K. I. Tsianos and M. G. Rabbat. Distributed consensus and optimization under communication delays. In *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pages 974–982, Sept 2011.
- [26] N. H. Vaidya, C. N. Hadjicostis, and A. D. Domínguez-García. Robust average consensus over packet dropping links: Analysis via coefficients of ergodicity. In *Proceedings of IEEE Conference on Decision and Control (CDC)*, pages 2761–2766, December 2012.

PLACE
PHOTO
HERE

Lili Su received the Bachelor degree in industrial engineering from Nankai University, Tianjin, China, in 2011 and M.S. degree in electrical and computer engineering from University of Illinois at Urbana-Champaign, USA. She is a Ph.D candidate at the department of electrical and computer engineering of University of Illinois at Urbana-Champaign. She is broadly interested in fault-tolerant computing, distributed algorithms, optimization, social computing, and security.

Nitin H. Vaidya received the B. S. degree in electrical and electronics engineering from Birla Institute of Technology and Science, Pilani, India, in 1986, M.E. degree in computer science and engineering from Indian Institute of Science-Bangalore in 1988, M.S. degree and Ph.D. degree in electrical and computer engineering from University of Massachusetts, Amherst in 1991 and 1993, respectively.

Since 2001, he has been with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, IL, USA.